



SWEN90016

# Software Processes & Project Management

Cost Estimation



Become familiar with

Agile

User Stories and Story Points and Velocity

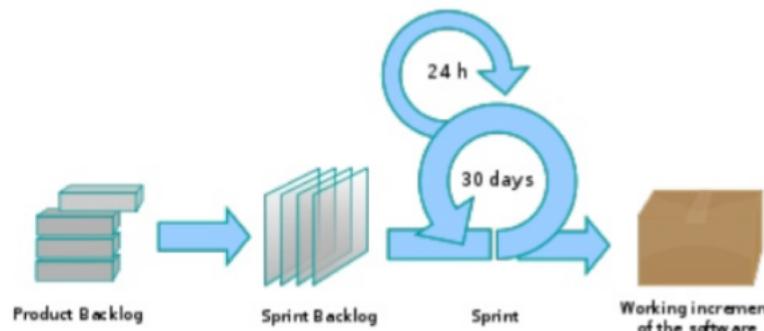
Formal

Function Point Analysis and COCOMO II

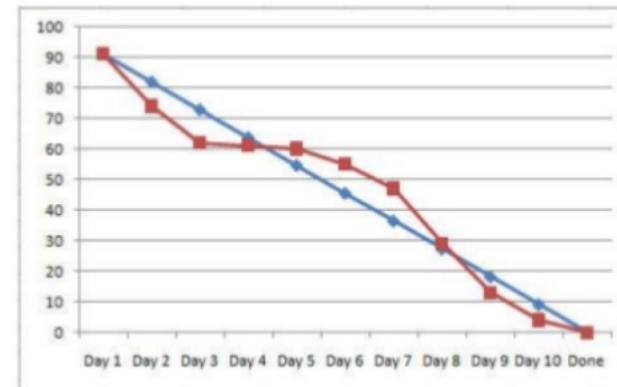


AVAILABILITY

From Lecture 2, slide 49



From Tutorial 3, slide 11



## Roles

Product Owner  
Scrum Master  
Development Team

## Ceremonies

Daily Stand Up  
Sprint Planning  
Sprint Review  
Sprint Retrospective

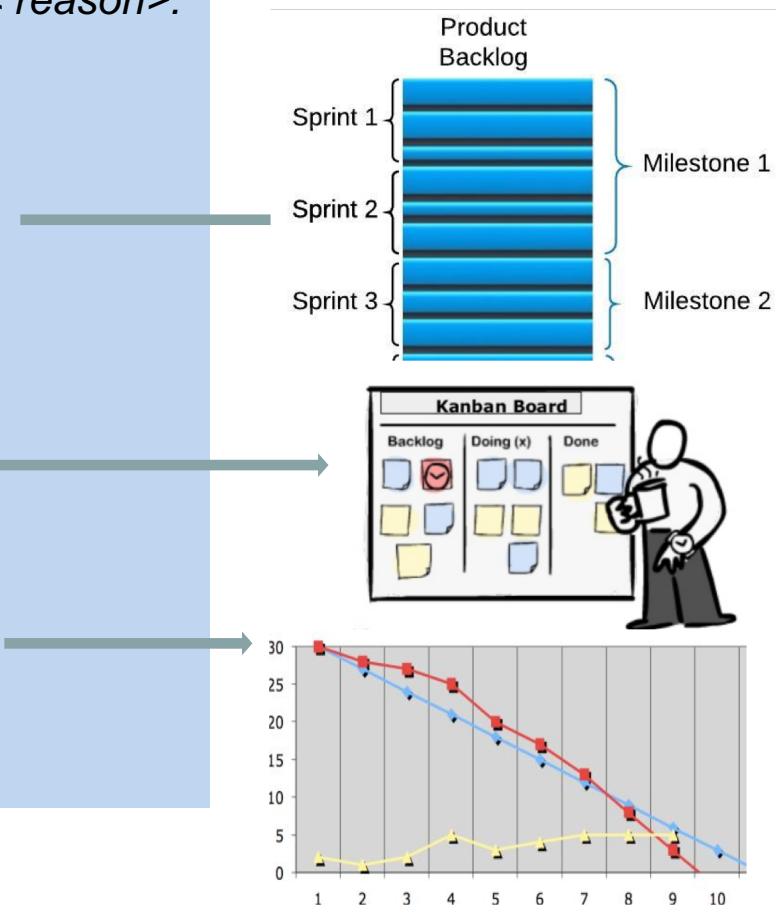
From Lecture 2, slide 51

## Artifacts

User Story  
Product Backlog  
Sprint Backlog  
Burndown Chart  
Burnup Chart

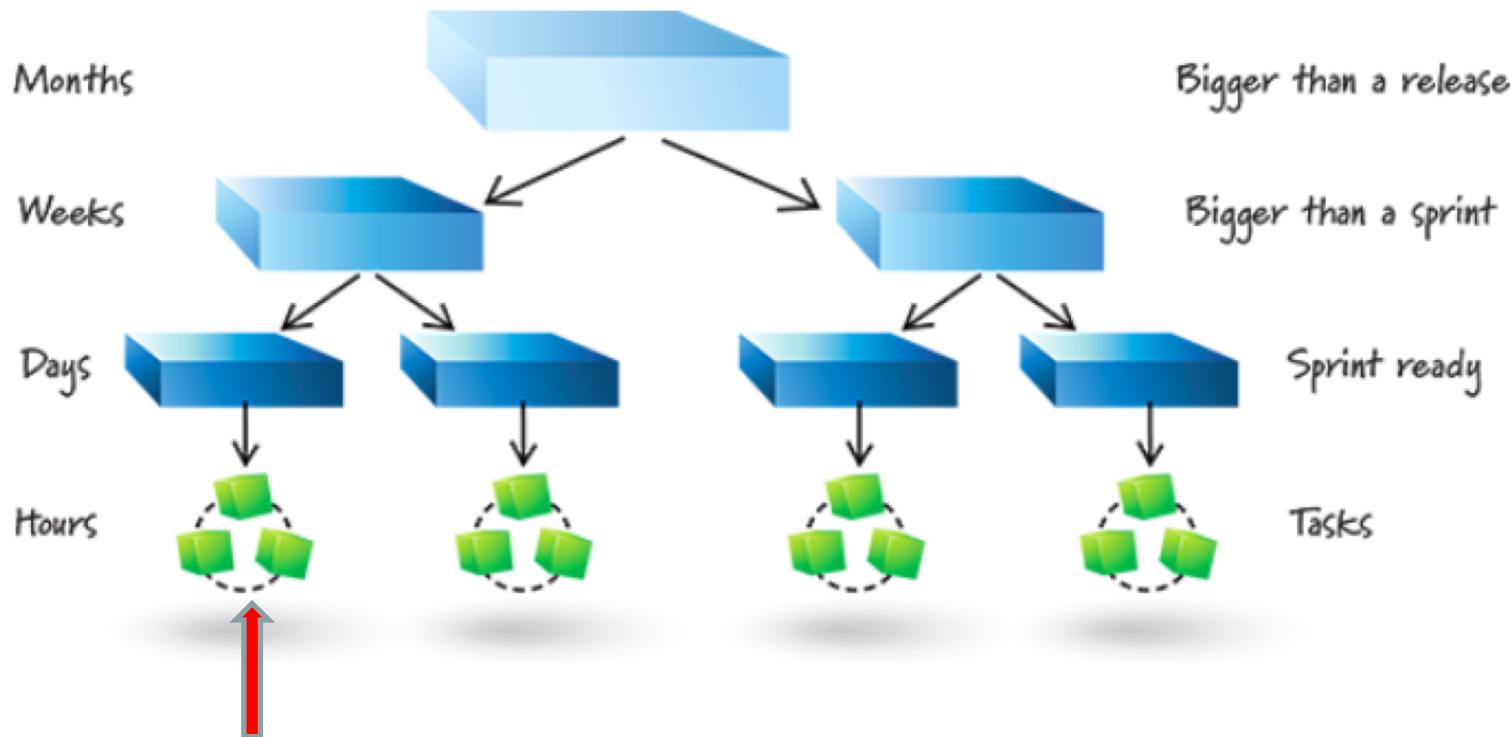


- User Stories
  - As a <user>, I want <goal> so that <reason>.
- Product Backlog
  - Features listed in client priority order
  - Release milestones annotated to list
- Sprint Backlog
  - Features selected for this iteration
  - Visual Kanban board
- Burn Down Chart
  - Measure the features **100% done**





From Tutorial 3, slide 9



Product Owner has a *conversation* with the Developer to understand requirement



AVAILABILITY

## (Sprint) User Story

- A developer's perspective
- A conversation placeholder

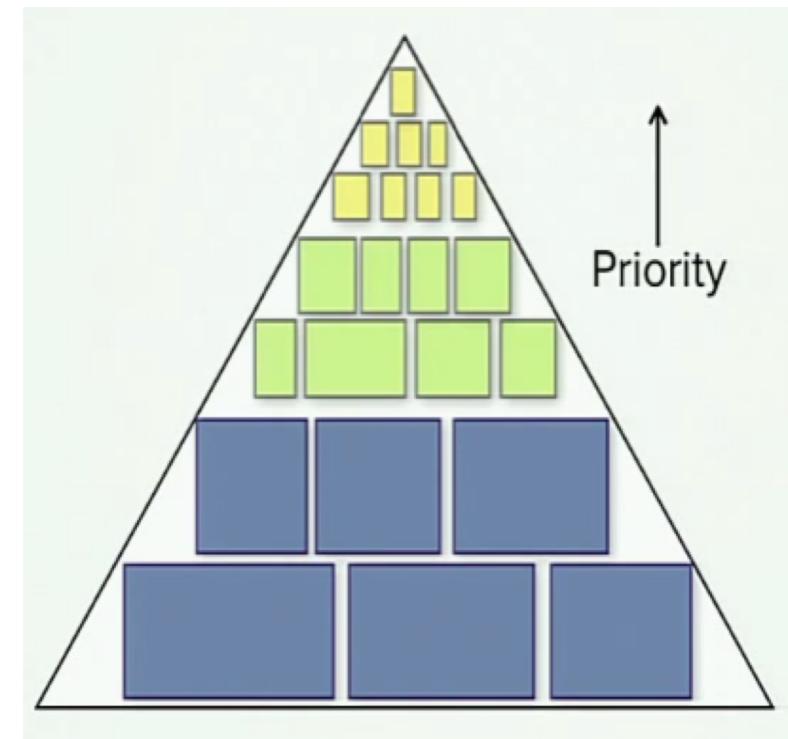
## Feature User Story

- Product capabilities
- Product Owner perspective

## Epic User Story

- New business services
- A product

Contentious! Advice from  
the internet may vary ...





**Story points:** a relative measure of the size of a user story  
(the requirements of the system are documented using user stories)

From Lecture 6, slide 71

raw values are unimportant

*relative values* matter

2 point story is twice as long  
as a 1 point story

limit range of Sprint Backlog  
estimates to 1-10





## A practical Example of Size vs Duration

- I am tasked with moving a large pile of dirt from the front of my home to the back yard.
- I could look at the pile of dirt, assess my tools [a shovel and a wheelbarrow], and directly estimate the job at two hours.
- In arriving at this estimate I bypassed any estimate of size and went directly to an estimate of duration.



## A practical Example of Size vs Duration

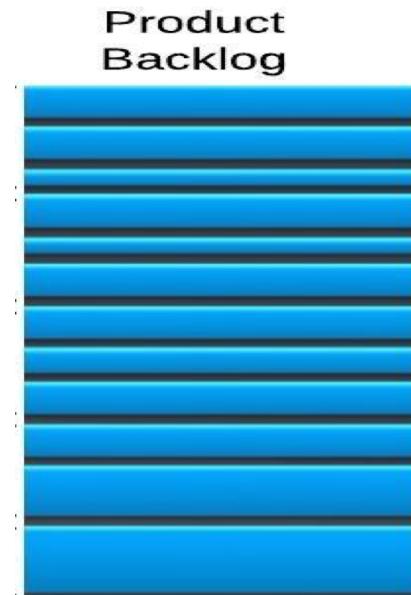
- Suppose instead that I look at the pile and estimate its size.
- Based on its dimensions I estimate the pile to contain about 100 cubic meters of dirt. This is my estimate of the size of this project.
- We want to know how long it will take to move the dirt: **Duration**
- We need to convert the estimate of size [100 cubic meters] into an estimate of duration.
- A label on my wheelbarrow says it has a capacity of two cubic meters.
- Dividing 100 cubic meters by 2 cubic meter, I decide that moving the dirt will take 50 trips with the wheelbarrow.
- I estimate that each trip will take three minutes to load the wheelbarrow, two minutes to walk to the back yard and dump the dirt, and one minute to walk back with the empty wheelbarrow. Total trip time will be six minutes.
- Since I anticipate making 50 trips taking 6 minutes each, my estimate of duration is 300 minutes or 5 hours.



Project:  
phase  
Initiation

Fixed Date and Time constraints

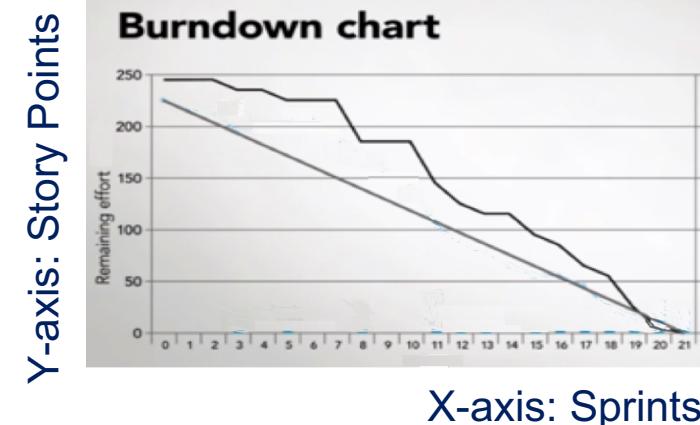
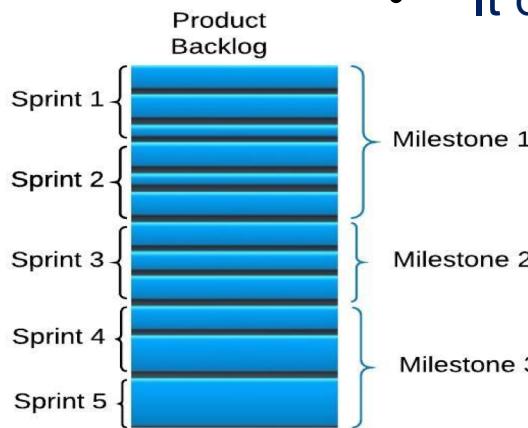
- Business Roadmap identifies candidate project
- Product vision established with external stakeholders
  - Create Product Backlog





Project phase:  
**Initial Sprint Planning**

- The groomed Product Backlog is estimated in Story Points
  - Cheap & quick estimation
  - Low quality indicators of {easy, medium hard}
  - Let estimates have larger values, like 21 or 100 are valid
- Find the dev team's Story-Point **Velocity** measure
  - It determines the **release** schedule





Project phase:  
every Sprint Planning

- Create Sprint Backlog      Fixed Date and Time constraints
  - Select high value User Stories from Product Backlog
  - Use velocity to fit appropriate number of Story Points
- Decompose selected User Stories on Sprint Backlog
  - Do Just-In-Time detailed estimation
  - Check number of Story Points will still fit
    - Detailed high quality estimation
    - Let estimates have smaller values, like 1 or 10 are valid

Humans have good judgement across one order of magnitude, but beyond that, humans are unreliable



Project phase:  
**every Sprint Planning**

Fixed Scope constraints

The User Stories can be decomposed into tasks,

- Optionally estimate tasks in hours

Less accurate [www.scruminc.com/story-points-why-are-they-better-than/](http://www.scruminc.com/story-points-why-are-they-better-than/)

- A full task level Sprint Backlog estimated in hours is equivalent to a formal schedule (Gantt)

More work [www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog](http://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog)

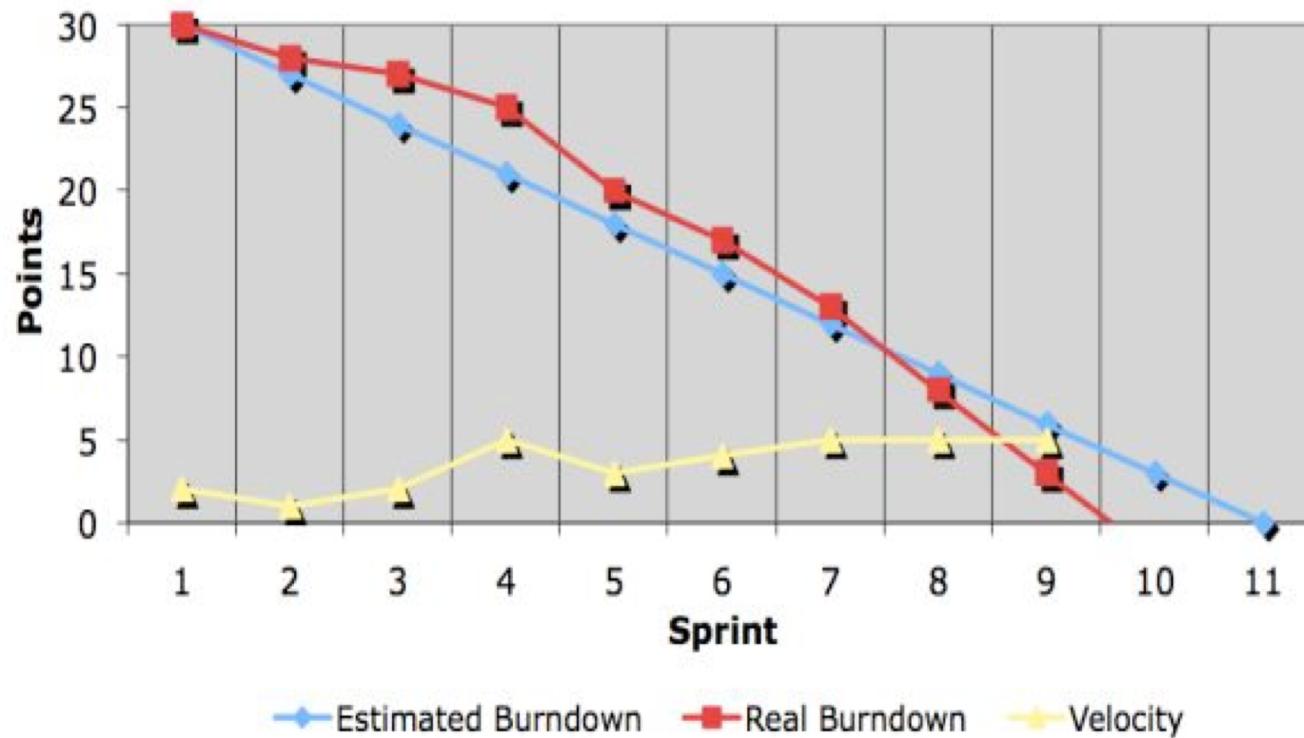


# Sprint Monitoring

SWEN90016 Software Processes and Project Management

Fixed Date and Time constraints

Project phase:  
Sprint



- Sprint Burn-down chart **monitors actual velocity**
- Scrum Master updated chart after daily standup

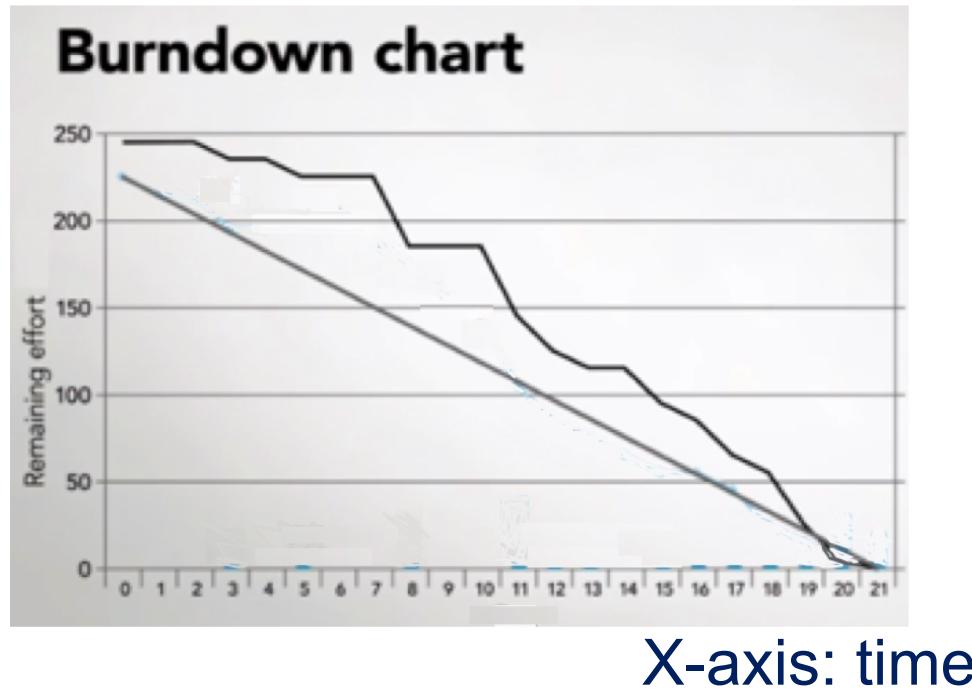


Fixed Date and Time constraints

Velocity determines the slope of the BurnDown charts

- The Scrum Master can track remaining effort
- Predict when the release milestones will be reached

Y-axis: effort



- Ideal schedule is the straight line
- Actual schedule is the jagged line
- The height of the chart shows the amount of work remaining



A project has this groomed **Product Backlog**, consisting of these **User Stories** which have been estimated to have these **Story Points**.

An established development team has an average **velocity** of **seven** User Story Points per fortnight.

Product Backlog	
User Story	Story Point
Story_1	3
Story_2	5
Story_3	13
Story_4	8
Story_5	1
Story_6	3
Story_7	2

1. Estimate how many weeks this team will take to deliver?
2. If the team actually completes the first two User Stories in two weeks, then what is the actual velocity of the team?
3. If a new User Story with Story Point=1 is added at the start of week 3, then in how many weeks do you estimate this project will take to be delivered now?



# Burn Down Chart: Activity

SWEN90016 Software Processes and Project Management

A project has this groomed **Product Backlog**, consisting of these **User Stories** which have been estimated to have these **Story Points**.

An established development team has an average **velocity** of **seven** User Story Points per fortnight.

Product Backlog	
User Story	Story Point
Story_1	3
Story_2	5
Story_3	13
Story_4	8
Story_5	1
Story_6	3
Story_7	2

4. Will User\_Story\_3 fit into a single sprint?
  
5. What process does Scrum have for completing User\_Story\_3 ?



## Top Down strategy:

Use cost of a previous similar project, size and effort  
Source Lines of Code, Function Points, Cocomo

## Bottom up strategy:

Estimate individual work items and sum  
WBS, Agile Story Points and Velocity

## Parametric:

use project characteristics in a mathematical model  
NVP, ROI, IRR



Become familiar with

Agile

User Stories and Story Points and Velocity

Formal

Function Point Analysis and COCOMO II

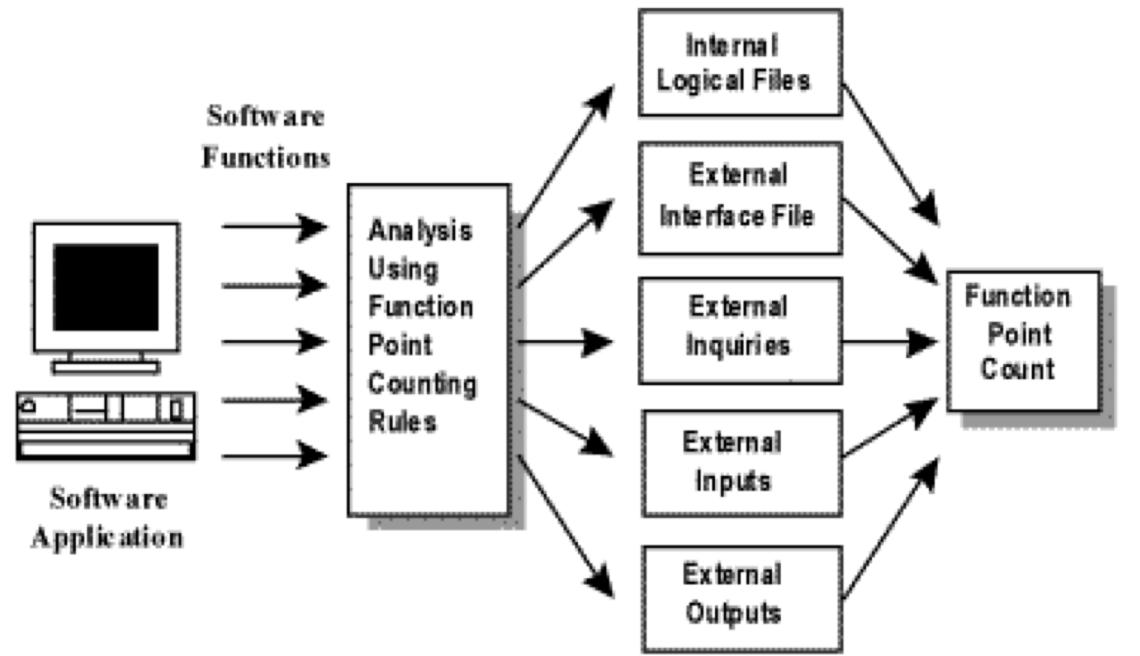


PMBOK

Historic Data

Done at any  
time in project  
lifecycle

## What are they?





# FP Computation Steps

1. Categorize functional requirements and count

Example: *Category* = {internal file, external file, input, output, query}

2. Estimate a *Complexity Level* for each category

*Complexity Level* = {simple, average, complex}

3. Compute *count total* of Function Points,  
(see next slide)

**Unadjusted Function Points** =  
sum (functions \* complexity value)

**Value Adjustment Factor** =  
apply expert opinion to your project estimates

4. Estimate *Value Adjustment Factors*

**Adjusted Function Points** =  
multiply business function by VAF

5. Compute *total function point count*



# FP Computation Steps

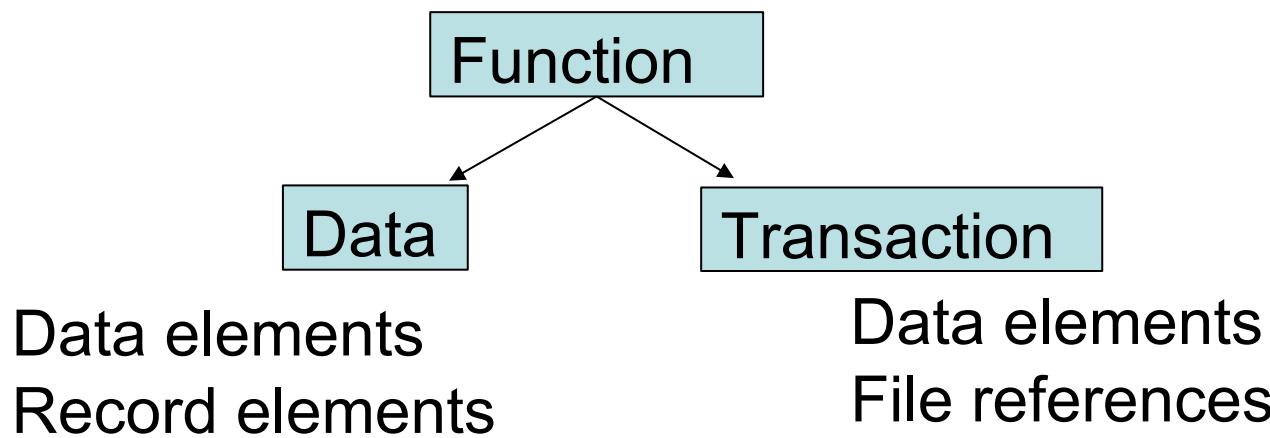
1. Categorize functional requirements and count

Example: *Category* = {internal file, external file, input, output, query}

2. Estimate a  
*Complexity Level*  
for each category

*Complexity Level* = {simple, average, complex}

Count functions from the Software Requirements Specification (SRS)





## Historic Data

complexity values

Category	Simple Function Count	Weight	Average Function Count	Weight	Complex Function Count	Weight	Sub total
Internal Logical File	5	3		4	2	6	
External Interface File`		4		5	1	7	
External Input	2	3		4		6	
External Output	5	7	2	10	2	15	
External Inquiries/Queries	2	5		7		10	
<b>Unadjusted Total</b>							

Factors published from 2,192 recent Function Point projects

<http://www.qsm.com/resources/function-point-languages-table>



Given the following business functions,  
how many *Unadjusted* Function Points exist?

Fill in the table.

Category	Simple Function Count	Weight	Average Function Count	Weight	Complex Function Count	Weight	Sub total
Internal Logical File	5	3		4	2	6	
External Interface File`		4		5	1	7	
External Input	2	3		4		6	
External Output	5	7	2	10	2	15	
External Inquiries/Queries	2	5		7		10	
<b>Unadjusted Total</b>							



The Constructive Cost Model:

Here is a playpen to try: <http://csse.usc.edu/tools/cocomoii.php>

Fill in the details for the Language Research Project.

Extra details to get started: let there be:

Sizing method: **135 Function Points**

The **Java** development language

The **cost per person-month** is **\$1500**



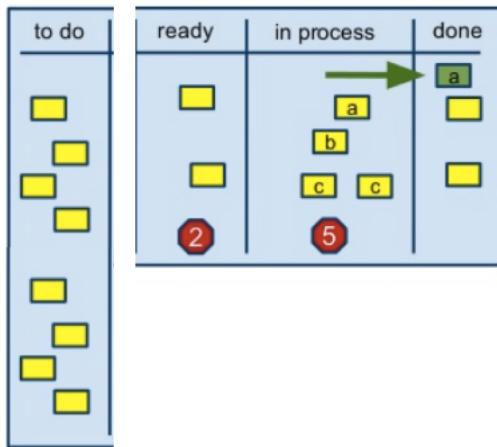
# Thank You!



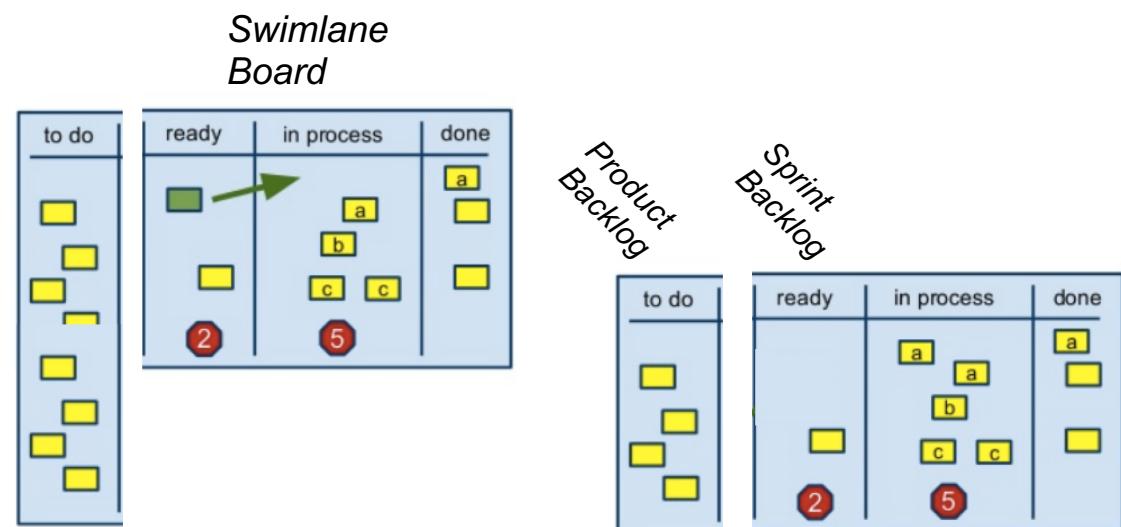
# Velocity revision

From Tutorial 6, slide 21

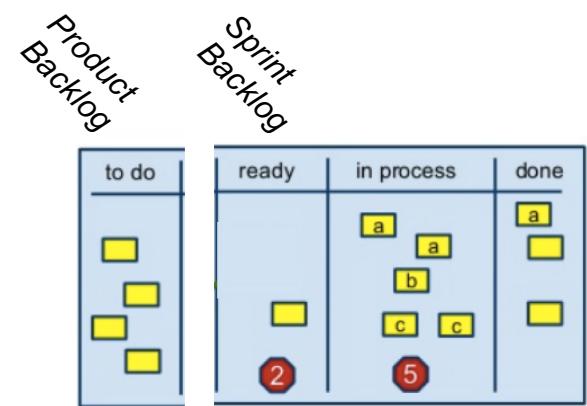
**Velocity:** velocity is the number of story points “done” by the Scrum development team in a Sprint time period.



Team member A completes code for a card and moves it to “done”



Team member A “pulls” a new card from “ready” and moves it to “doing”



The Product Owner selects the next priority set of cards (Sprint Backlog) and moves it to “ready”



Description of how to estimate initial velocity

[www.mountaingoatsoftware.com/blog/how-to-estimate-velocity-as-an-agile-consultant](http://www.mountaingoatsoftware.com/blog/how-to-estimate-velocity-as-an-agile-consultant)

Description of User Stories which gave been decomposed into tasks and estimated in hours,  
This optional slows down the productivity of the team,

[www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog](http://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog)

Research which assets Story Points are more accurate than hours,

[www.scruminc.com/story-points-why-are-they-better-than/](http://www.scruminc.com/story-points-why-are-they-better-than/)

“Story Points are considered faster, better, and cheaper than hours and the highest performing teams completely abandon any hourly estimation as they view it as waste that just slows them down.”